

Guía definitiva: cuadro de mando de sensores en iPhone con permiso de movimiento WebKit

11:53



SENSOR DASHBOARD



ACTIVAR SENSORES DE
MOVIMIENTO



NIVEL DE BATERÍA

Nivel: 60%

Cargando: No



ACELERÓMETRO (MOVIMIENTO)

X: 0.11 m/s²

Y: 0.00 m/s²

Z: 0.35 m/s²



ORIENTACIÓN DEL DISPOSITIVO

Alpha (Z): 356°

Beta (X): 79°

Gamma (Y): 1°



CONECTIVIDAD

Estado: Conectado (Online)

Modo: Navegador WebKit



MONITOR DE PANTALLA

Resolución: 1170 x 2532 Pixels

Relación de Aspecto: 3

En las versiones actuales de iOS, Apple protege los sensores de movimiento frente al rastreo web. Aunque el acelerómetro y el giroscopio no necesitan una declaración en el archivo Info.plist de Xcode, **requieren obligatoriamente una interacción del usuario (un clic) y una confirmación en pantalla** para empezar a emitir datos.

Aquí tienes el tutorial completo y el código definitivo para crear un Dashboard funcional.

Paso 1: Crear el directorio y la estructura del proyecto

Abre la **Terminal** de tu Mac y ejecuta los siguientes comandos para inicializar el entorno desde cero:

```
# 1. Crear la carpeta del proyecto
mkdir dashboardsensores
```

```
# 2. Entrar a la carpeta
cd dashboardsensores
```

```
# 3. Inicializar el proyecto Node.js
npm init -y
```

```
# 4. Instalar las dependencias principales de Capacitor
npm install @capacitor/core @capacitor/cli
```

Paso 2: Inicializar la configuración de la App

Configura la identidad de la aplicación ejecutando en la terminal:

```
npx cap init "SensorDash" "com.tuusuario.sensordash" --web-dir=www
```

Paso 3: Instalar el Plugin de Dispositivo (Device)

Instalamos el plugin oficial para capturar los datos de hardware de la batería de forma nativa:

```
npm install @capacitor/device
```

Paso 4: Crear la interfaz del Cuadro de Mando (HTML5 Completo)

Creamos la carpeta web donde alojaremos nuestro diseño:

```
mkdir www
```

Crea el archivo **index.html** dentro de la carpeta **www** y pega este código unificado que ya incluye el botón de activación obligatorio para iOS:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Sensor Dashboard iPhone</title>
  <style>
    body {
      background-color: #0b0f19;
      color: #f3f4f6;
      margin: 0;
      padding: 20px;
    }
    h1 {
      text-align: center;
      color: #4f46e5;
      font-size: 26px;
      margin-bottom: 20px;
      text-transform: uppercase;
```

```
        letter-spacing: 1px;
    }
    .btn-container {
        text-align: center;
        margin-bottom: 25px;
    }
    .btn-sensor {
        background-color: #4f46e5;
        color: white;
        border: none;
        padding: 14px 28px;
        font-size: 16px;
        font-weight: bold;
        border-radius: 12px;
        cursor: pointer;
        box-shadow: 0 4px 12px rgba(79, 70, 229, 0.4);
        transition: transform 0.1s ease;
    }
    .btn-sensor:active {
        transform: scale(0.95);
    }
    .grid {
        display: grid;
        grid-template-columns: 1fr;
        gap: 15px;
        width: 100%;
        max-width: 500px;
        margin: 0 auto;
    }
    .card {
        background: #111827;
        border: 1px solid #1f2937;
        border-radius: 16px;
        padding: 15px;
        box-shadow: 0 4px 10px rgba(0,0,0,0.3);
    }
    .card-title {
        font-size: 14px;
        color: #9ca3af;
        text-transform: uppercase;
        font-weight: bold;
    }
```

```

        margin-bottom: 10px;
        border-bottom: 1px solid #1f2937;
        padding-bottom: 5px;
    }
    .sensor-data {
        font-size: 16px;
        color: #10b981;
    }
    .error {
        color: #ef4444 !important;
    }
</style>
</head>
<body>

    <h1>Sensor Dashboard</h1>

    <div class="btn-container">
        <button class="btn-sensor"
onclick="activarSensores()">▶ ACTIVAR  SENSORES  DE
MOVIMIENTO</button>
    </div>

    <div class="grid">

        <div class="card">
            <div class="card-title">▶ Nivel de Batería</div>
            <div class="sensor-data">
                Nivel: <span id="bat-level">--%</span><br>
                Cargando: <span id="bat-charging">--</span>
            </div>
        </div>

        <div class="card">
            <div class="card-title">▶ Acelerómetro
(Movimiento)</div>
            <div class="sensor-data">
                X: <span id="acc-x">0.00</span> m/s2<br>
                Y: <span id="acc-y">0.00</span> m/s2<br>
                Z: <span id="acc-z">0.00</span> m/s2
            </div>
        </div>
    </div>

```

```

</div>

<div class="card">
    <div class="card-title">[] Orientación del
Dispositivo</div>
    <div class="sensor-data">
        Alpha (Z): <span id="ori-alpha">0°</span><br>
        Beta (X): <span id="ori-beta">0°</span><br>
        Gamma (Y): <span id="ori-gamma">0°</span>
    </div>
</div>

<div class="card">
    <div class="card-title">[] Conectividad</div>
    <div class="sensor-data">
        Estado: <span id="net-
status">Detectando...</span><br>
        Modo: <span id="net-type">Navegador
WebKit</span>
    </div>
</div>

<div class="card">
    <div class="card-title">[] Monitor de
Pantalla</div>
    <div class="sensor-data">
        Resolución: <span id="screen-
res">0x0</span><br>
        Relación de Aspecto: <span id="screen-
pixel">0</span>
    </div>
</div>

</div>

<script>
    // --- 1. CONFIGURACIÓN DE BATERÍA (CAPACITOR) ---
    async function initBattery() {
        try {
            if (window.Capacitor &&
window.Capacitor.Plugins.Device) {

```

```

        const info = await
window.Capacitor.Plugins.Device.getBatteryInfo();
        document.getElementById('bat-
level').textContent = Math.round(info.batteryLevel * 100) +
'%';
        document.getElementById('bat-
charging').textContent = info.isCharging ? "Sí <" : "No";
    }
} catch (e) { console.error(e); }
}

// --- 2. SOLICITUD DE PERMISOS DE MOVIMIENTO
(WEBKIT/iOS) ---
    async function activarSensores() {
        // Verificar si el motor WebKit de iOS requiere
solicitar el permiso mediante software
        if (typeof DeviceOrientationEvent !== 'undefined'
&& typeof DeviceOrientationEvent.requestPermission ===
'function') {
            try {
                const permissionState = await
DeviceOrientationEvent.requestPermission();
                if (permissionState === 'granted') {
                    iniciarEscuchaSensores();
                } else {
                    alert("❌ Permiso denegado para acceder
a los sensores de movimiento.");
                }
            } catch (error) {
                console.error(error);
                alert("Error al activar: Asegúrate de
pulsar el botón directamente.");
            }
        } else {
            // Navegadores que no restringen el acceso por
defecto (Android o iOS antiguos)
            iniciarEscuchaSensores();
        }
    }

    function iniciarEscuchaSensores() {

```

```

        // Escuchar eventos del Acelerómetro
        window.addEventListener('devicemotion', (event) =>
        {
            if (event.acceleration) {
                document.getElementById('acc-
x').textContent = (event.acceleration.x || 0).toFixed(2);
                document.getElementById('acc-
y').textContent = (event.acceleration.y || 0).toFixed(2);
                document.getElementById('acc-
z').textContent = (event.acceleration.z || 0).toFixed(2);
            }
        });

        // Escuchar eventos de la Orientación espacial
        (Giroscopio)
        window.addEventListener('deviceorientation',
        (event) => {
            document.getElementById('ori-
alpha').textContent = Math.round(event.alpha || 0) + '°';
            document.getElementById('ori-
beta').textContent = Math.round(event.beta || 0) + '°';
            document.getElementById('ori-
gamma').textContent = Math.round(event.gamma || 0) + '°';
        });
    }

    // --- 3. DETECCIÓN DE RED ---
    function updateNetworkStatus() {
        const status = navigator.onLine ? "Conectado
(Online)" : "Desconectado (Offline)";
        const element = document.getElementById('net-
status');
        element.textContent = status;
        element.className = navigator.onLine ? "" :
"error";
    }

    window.addEventListener('online',
updateNetworkStatus);
    window.addEventListener('offline',
updateNetworkStatus);

```

```

// --- 4. DETECCIÓN DE PANTALLA ---
function getScreenData() {
    const width = window.screen.width *
window.devicePixelRatio;
    const height = window.screen.height *
window.devicePixelRatio;
    document.getElementById('screen-res').textContent
= ${width} x ${height} Pixels;
    document.getElementById('screen-
pixel').textContent = window.devicePixelRatio;
}

// --- INICIALIZACIÓN GLOBAL ---
window.addEventListener('DOMContentLoaded', () => {
    getScreenData();
    updateNetworkStatus();
    setInterval(initBattery, 3000);
    setTimeout(initBattery, 500);
});
</script>
</body>
</html>

```

Paso 5: Sincronizar y Ejecutar en Xcode

De vuelta en la **Terminal**, sincroniza la nueva interfaz:

```

npm install @capacitor/ios
npx cap add ios
npx cap sync ios

```

1. Abre tu Xcode: `npx cap open ios`.
2. Pon el iPhone en **Modo Avión** (con Wi-Fi y Bluetooth apagados manualmente) y conéctalo por cable USB.
3. Elige tu iPhone real en el selector de arriba, limpia la caché con **Product -> Clean Build Folder** y pulsa en **Play**.

Cómo probarlo en el iPhone:

Cuando se abra la aplicación, verás los sensores a cero. Pulsa el botón «**ACTIVAR SENSORES DE MOVIMIENTO**», acepta la ventana emergente que te muestra iOS por pantalla, ¡y listo! Verás los valores variar en tiempo real al mover tu teléfono. ¡Un post perfecto y técnicamente impecable!